

УПРАВЛІННЯ В ОРГАНІЗАЦІЙНИХ СИСТЕМАХ

УПРАВЛЕНИЕ В ОРГАНИЗАЦИОННЫХ СИСТЕМАХ

MANAGEMENT IN ORGANIZATIONAL SYSTEMS

UDC 519.854.2

DOI: 10.20998/2079-0023.2019.01.04

A. A. PAVLOV, E. B. MISURA, O. V. MELNIKOV

TOTAL WEIGHTED TARDINESS MINIMIZATION FOR TASKS WITH A COMMON DUE DATE ON PARALLEL MACHINES IN CASE OF AGREEABLE WEIGHTS AND PROCESSING TIMES

We consider n tasks scheduling problem on m identical parallel machines by the criterion of minimizing the total weighted tardiness of tasks. All tasks arrive for processing at the same time. Weights and processing times are agreeable, that is, a greater weight of a task corresponds to a shorter processing time. In addition, we have arbitrary start times of machines for tasks processing. The times may be less or greater than the due date or to coincide with it. The problem in this formulation is addressed for the first time. It can be used to provide planning and decision making in systems with a network representation of technological processes and limited resources. We give efficient PSC-algorithm with $O(mn \log n)$ complexity that includes the polynomial component and the approximation algorithm based on permutations of tasks. The polynomial component contains sufficient signs of optimality of the obtained solutions and allows to obtain an exact solution by polynomial subalgorithm. In the case when the sufficient signs of optimality do not fulfill, we obtain approximate solution with an estimate of deviation from the optimum for each individual problem instance of any practical dimension. We show that a schedule obtained as a result of the problem solving can be split into two schedules: the schedule on machines which start time is less than or equal to the due date, and the schedule on machines which start after the due date. Optimization is only done in the first schedule. The second schedule is optimal by construction. Statistical studies of the PSC-algorithm showed its high efficiency. We solved problems with dimensions up to 40,000 tasks and up to 30 machines. The average time to solve the problem by the algorithm using the most efficient types of permutations was 27.3 ms for this dimension. The average frequency of an optimal solution obtaining amounted to 90.3 %. The average deviation from an optimum was no more than 0.000251.

Keywords: scheduling theory, parallel machines, total weighted tardiness, common due date, agreeable weights, PSC-algorithm

O. A. ПАВЛОВ, О. Б. МИСЮРА, О. В. МЕЛЬНИКОВ

МІНІМІЗАЦІЯ СУМАРНОГО ЗВАЖЕНОГО ЗАПІЗНЕННЯ ЗАВДАНЬ ІЗ СПІЛЬНИМ ДИРЕКТИВНИМ СТРОКОМ НА ПАРАЛЕЛЬНИХ ПРИЛАДАХ ДЛЯ ВИПАДКУ УЗГОДЖЕНИХ ВАГ І ТРИВАЛОСТЕЙ

Розглядається задача складання розкладів виконання n завдань на m ідентичних паралельних приладах за критерієм мінімізації сумарного зваженого запізнення завдань. Всі завдання надходять на обслуговування одночасно. Ваги і тривалості узгоджені, тобто завданню з меншою тривалістю відповідає більша вага. Додатково, задані довільні моменти початку роботи приладів на виконання завдань, які можуть бути як менше директивного строку, так і більше, або співпадати з ним. У такій постановці задача розв'язується вперше. Вона може використовуватися для забезпечення планування та прийняття рішень в системах з мережним представленням технологічних процесів та обмеженими ресурсами. Наведено ефективний ПДС-алгоритм її розв'язання із трудомісткістю $O(mn \log n)$, який включає поліноміальну складову з достатніми ознаками оптимальності одержуваних розв'язків, яка дозволяє отримувати точний розв'язок поліноміальним підалгоритмом. У разі невиконання достатніх ознак оптимальності ми отримуємо наближений розв'язок з оцінкою відхилення отриманого розв'язку від оптимального для кожної індивідуальної задачі будь-якої практичної розмірності. Показано, що розклад, отриманий в результаті розв'язання задачі, можна умовно розбити на два розклади – розклад на приладах, момент початку роботи яких менше або дорівнює директивному строку, та розклад на приладах, що починають роботу після директивного строку. Оптимізація виконується тільки у першому розкладі. Другий розклад оптимальний за побудовою. Статистичні дослідження ПДС-алгоритму показали його високу ефективність. Розв'язувались задачі з розмірністю до 40 000 завдань з числом приладів до 30. Середній час розв'язання задачі алгоритмом, що використовує найбільш ефективні типи перестановок, склав 27,3 мс при цій розмірності. Середня частота отримання оптимального розв'язку склала до 90,3 %. Середнє відхилення від оптимуму – не більш, ніж 0,000251.

Ключові слова: комбінаторна оптимізація, теорія розкладів, паралельні прилади, сумарне зважене запізнення, спільний директивний строк, узгоджені ваги, ПДС-алгоритм

A. A. ПАВЛОВ, Е. Б. МИСЮРА, О. В. МЕЛЬНИКОВ

МИНИМИЗАЦИЯ СУММАРНОГО ВЗВЕШЕННОГО ЗАПАЗДЫВАНИЯ ЗАДАНИЙ С ОБЩИМ ДИРЕКТИВНЫМ СРОКОМ НА ПАРАЛЛЕЛЬНЫХ ПРИБОРАХ ДЛЯ СЛУЧАЯ СОГЛАСОВАННЫХ ВЕСОВ И ДЛИТЕЛЬНОСТЕЙ

Рассматривается задача составления расписаний выполнения n заданий на m идентичных параллельных приборах по критерию минимизации суммарного взвешенного опоздания заданий. Все задания поступают на обслуживание одновременно. Веса и длительности согласованы, то есть заданию с меньшей длительностью соответствует больший вес. Дополнительно, заданы произвольные моменты начала работы приборов

© A. A. Pavlov, E. B. Misura, O. V. Melnikov, 2019

на выполнение заданий, которые могут быть как меньше директивного срока, так и больше, либо совпадать с ним. В такой постановке задача решается впервые. Она может использоваться для обеспечения планирования и принятия решений в системах с сетевым представлением технологических процессов и ограниченными ресурсами. Приведен эффективный ПДС-алгоритм ее решения с трудоемкостью $O(mn \log n)$, включающий полиномиальную составляющую с достаточными признаками оптимальности получаемых решений, которая позволяет получать точное решение полиномиальным подалгоритмом. В случае невыполнения достаточных признаков оптимальности мы получаем приближенное решение с оценкой отклонения полученного решения от оптимального для каждой индивидуальной задачи любой практической размерности. Показано, что расписание, полученное в результате решения задачи, можно условно разбить на два расписания – расписание на приборах, момент начала работы которых меньше или равен директивному сроку, и расписание на приборах, начинающих работу после директивного срока. Оптимизация выполняется только в первом расписании. Второе расписание оптимально по построению. Статистические исследования ПДС-алгоритма показали его высокую эффективность. Решались задачи с размерностью до 40 000 заданий с числом приборов до 30. Среднее время решения задачи алгоритмом, использующим наиболее эффективные типы перестановок, составило 27,3 мс при этой размерности. Средняя частота получения оптимального решения составила до 90,3 %. Среднее отклонение от оптимума – не более 0,000251.

Ключевые слова: комбинаторная оптимизация, теория расписаний, параллельные приборы, суммарное взвешенное запаздывание, общий директивный срок, согласованные веса, ПДС-алгоритм

Introduction. Scheduling theory problems play an important role in calendar and operational planning of discrete type manufactures. In particular, those manufactures include small-series productions, aircraft and ship-building enterprises. Almost all known scheduling problems are NP-hard problems of combinatorial optimization. Therefore, creating for them efficient approximation algorithms is very important to solve practical problems of large dimensions.

The interest for tardiness criterion is due to its practical effect in the real life [1]. This criterion is among the most interesting criteria for production systems, especially in the current situation where competitiveness is becoming more and more intensive. Suppliers do ensure their markets and customers. For that, they must have a high service quality while focusing on delivery dates.

We solve in this paper the problem of the total weighted tardiness minimization on identical parallel machines with a common due date and agreeable weights (WTPA). It can be used to provide planning and decision making in systems with a network representation of technological processes and limited resources. In particular, it is used in the algorithmic ware of the four-level model of planning (including operative planning) and decision making [2].

WTPA problem statement. Given a set of tasks $J = \{j_1, j_2, \dots, j_n\}$, the number of machines is m . For each task $j \in J$, we know its processing time l_j and the weight ω_j . The weights are agreeable: if $l_i < l_j$, then $\omega_i \geq \omega_j$. All tasks have the same due date d . We need to build a schedule σ of tasks $j \in J$ on m machines that minimizes the function:

$$F(\sigma) = \sum_{j \in J} \omega_j \max[0, C_j(\sigma) - d], \quad (1)$$

where $C_j(\sigma)$ is the completion time of a task j in a schedule σ .

We assume that all tasks of the set J arrive simultaneously. The start times of the machines' operation for the processing of tasks $r_i \geq d$, $i = \overline{1, m}$, may be different. Here, $\overline{a, b}$ denotes the interval of integer numbers from a to b . Machines' idle times and interruptions in a task's processing are not allowed.

Literature review. This problem belongs to the class of NP-hard problems since it is NP-hard already for $m = 1$ and $r_1 = 0$ [3, 4].

Many modern methods of scheduling are described in the book [5]. Unified heuristics and annotated bibliography for a large class of scheduling problems with tardiness criteria are presented in [6]. The case of the problem with equal start times of machines and equal weights of tasks (TTP problem) was investigated in [2, 7–9]. A literature review for TTP problem in different formulations is given in [10]; this problem is also considered there for the case of different release dates of tasks. The class of schedules that contain an optimal solution is defined in [7], also an estimate of the deviation of the obtained solution from the optimum for each individual problem instance was formulated there. Kovalyov et al. [8] show that there is no polynomial approximation algorithm with a guaranteed relative error of a solution for TTP problem unless $P = NP$. Lawler et al. [11] indicated that the total tardiness problem on a single machine with a common due date is solved in $O(n^2)$ time. Its version with job weights is NP-hard in the ordinary sense [12].

Books [2, 9] present efficient PSC-algorithm to solve TTP problem, prove sufficient signs of optimality of the obtained solutions, and clarify the estimate of deviation of an obtained solution from the optimum. In contrast to the well-known estimate [7] which can be arbitrarily large even at an optimal solution, the estimate from [2, 9] is adequate: it is limited from above and from the bottom and shows the maximum possible reduction of the functional during an optimal solution construction. It was shown in [2, 9] that the PSC-algorithm for the problem solving, sufficient signs of optimality of the solutions obtained, and the estimate of deviation of a solution from the optimum for each individual problem instance obtained for TTP problem, are also valid for WTPA problem in the case of equal weights.

The above review shows that the problem under consideration is not presented in the scheduling literature in original formulation. This explains why we address it here. The problem in above formulation is addressed for the first time. The purpose of this paper is to develop the PSC-algorithm for the WTPA problem solving with an estimate of deviation of an obtained solution from the optimum for each individual problem instance.

Theoretical foundations. We now present basics of PSC-algorithm for TTP problem [2, 9, 10] which underlies the PSC-algorithm for WTPA problem.

Algorithm for initial schedule construction.

1. Renumber the tasks $j \in J$ in non-decreasing order of their processing times.

2. Set *release time* (completion time of all assigned jobs) of all machines to zero: $C_j = 0 \quad \forall i = \overline{1, m}$.
3. Select the next unassigned task j with the minimum processing time l_j . Assign it to a machine i with the minimum release time C_i .
4. Determine new release time of the machine i : $C_i = C_i + l_j$.
5. If all tasks were assigned, the algorithm terminates. Otherwise, go to step 3.

As a result of the above algorithm's execution, tasks $j \in J$ are split into three subsets [2]:

$P_i(\sigma)$ is the set of non-tardy tasks in the schedule of machine i ;

$S_i(\sigma)$ is the set of "straddling" tardy tasks in the schedule of machine i for which

$$C_j - l_j < d, C_j > d, \quad \forall j \in S_i(\sigma),$$

where $C_j - l_j$ is the start time of a task j ;

$Q_i(\sigma)$ is the set of "fully" tardy tasks in the schedule of machine i for which $C_j - l_j \geq d, \forall j \in Q_i(\sigma)$.

We use the notation from [2]:

$$P = \bigcup_{i=\overline{1, m}} P_i; S = \bigcup_{i=\overline{1, m}} S_i; Q = \bigcup_{i=\overline{1, m}} Q_i;$$

$R_i(\sigma)$ the time reserve of machine i in a schedule σ : $R_i(\sigma) = d - \sum_{j \in P_i(\sigma)} l_j$; $R_\Sigma(\sigma) = \sum_{i=1}^m R_i(\sigma)$;

$\Delta_i(\sigma)$ the tardiness of straddling task $j \in S_i(\sigma)$ in regard to the due date: $\Delta_i(\sigma) = \sum_{j \in P_i(\sigma) \cup S_i(\sigma)} l_j - d$;

$$\Delta_\Sigma(\sigma) = \sum_{i=1}^m \Delta_i(\sigma).$$

Theorem 1 [7]. There is an optimal schedule that satisfies the conditions:

- 1) $P \cup S = \{1, 2, \dots, |P \cup S|\}$;
- 2) if $|P \cup S| < n$, then $\sum_{j \in P_i \cup S_i} l_j \geq d$ and Q_i contains those and only those elements which differ from $|P \cup S| + i$ by an amount which is a multiple of m , $i = \overline{1, m}$.

Corollary 1 [7]. Suppose that the tasks processing on L -th machine cannot be started before the time point $r_L \geq 0, L = \overline{1, m}$. Schedule σ where each successive task $k = 1, 2, \dots, n$ is assigned to be processed on the machine released before the others, corresponds to the smallest sum of completion times for all tasks.

Statement 1 [7]. When constructing an optimal schedule as a result of directed permutations, tasks can be moved only between sets P and S .

Let ψ_{PS} denote a class of schedules that correspond to the conditions of Theorem 1. $\psi_P \subset \psi_{PS}$ is a class of schedules satisfying the following additional conditions:

- 1) $P = \{1, 2, \dots, |P|\}$;
- 2) $\min_{j \in S_i(\sigma)} l_j > \max_{i=\overline{1, m}} R_i(\sigma)$;
- 3) if $l_{j_k} \leq l_{j_l}$, then $C_{j_k} - l_{j_k} \leq C_{j_l} - l_{j_l} \quad \forall j_k, j_l \in S(\sigma)$.

The number of tardy tasks on machines differs by a maximum of one in the class ψ_P [2]. We determine $R_i(\sigma)$ on machines with a smaller number of tardy tasks and $\Delta_i(\sigma)$ on those with a larger number of tardy tasks.

Two sufficient signs of optimality of a feasible solution were proved in [2]:

1) a schedule $\sigma \in \psi_P$ with the same number of tardy tasks on all machines (an even schedule) is optimal;

2) if $\Omega_\Sigma(\sigma) = \min(R_\Sigma(\sigma), \Delta_\Sigma(\sigma)) = 0$ in a schedule $\sigma \in \psi_P$, then the schedule σ is optimal.

The PSC-algorithm from [2] is based on directed permutations that decrease $\Delta_\Sigma(\sigma)$ on machines with a larger number of tardy tasks due to reserves $R_\Sigma(\sigma)$ on machines with a smaller number of tardy tasks or to build an even schedule. Let $\psi(\sigma_P) \subset \psi_{PS}$ denote the class of such schedules.

Theorem 2 [2]. The following estimate of deviation of the functional value from the optimum is valid for any schedule $\sigma \in \psi(\sigma_P)$: $f(\sigma) - f(\sigma^*) \leq \Omega_\Sigma(\sigma)$.

The case of the TTP problem, in which the start times of machines are less than the common due date, was considered in [2], and a PSC-algorithm was given. Let us consider the general case.

Study of WTPA problem properties. Let us call ω_j/l_j the priority of a task j . Since we have different start times of machines in the WTPA problem, in contrast to the TTP problem, let us give a new algorithm for construction of initial schedule $\sigma^{\text{ord}} \in \psi_P$.

Algorithm A0.

1. Renumber the tasks $j \in J$ in non-increasing order of priorities ω_j/l_j .
2. Renumber the machines in non-decreasing order of start times r_i .
3. Set the initial release times of machines: $C_i = r_i \quad \forall i = \overline{1, m}$.
4. Select an unassigned task j with the maximum priority. Assign it to a machine i with the minimum release time C_i .
5. Determine the new release time of the machine i : $C_i = C_i + l_j$.
6. If all tasks were assigned, the algorithm terminates. Otherwise, go to step 4.

Denote the obtained schedule as σ^{ord} (sigma ordered). Let us split σ^{ord} conditionally into σ^1 and σ^2 where σ^1 is the schedule of tasks on machines with $r_i < d$ and σ^2 is the schedule of tasks on machines with $r_i \geq d$.

Statement 2. The number of tardy tasks on machines in the schedule σ^1 differs by a maximum of one.

Statement 3. The number of tardy tasks on each of the machines with $r_i < d$ is greater than or equal to the number of tardy tasks on each of the machines with $r_i \geq d$.

Validity of Statements 2 and 3 is based on the algorithm for the schedule σ^{ord} construction.

The schedule σ^1 meets the requirements to the class ψ_P . The following theorem is true.

Theorem 3. The PSC-algorithm for the problem solving, sufficient signs of optimality of obtained solutions, and the estimate of deviation of a solution from the optimum for each individual problem instance which were obtained for TTP problem, are also valid for WTPA problem.

Proof. Initial schedule of the class ψ_P is constructed in TTP problem by distributing the list of tasks in non-decreasing order of their processing times, each to the machine with a minimum release time. Similarly, initial schedule in WTPA problem is built by distributing the

priority-ordered sequence of tasks, each to the machine with the minimum release time. But the priority-ordered sequence of tasks coincides with the sequence of tasks ordered by non-decreasing of their processing times, due to the agreeability of weights and processing times. Hence, the initial schedules in TTP and WTPA problems coincide. The PSC-algorithm is based on task permutations from machines with a greater number of tardy tasks to machines with a smaller number of tardy tasks. Thus, the task weights do not affect the implementation of the algorithm and are taken into account only when determining the functional value. This proves the theorem.

Theorem 4. There are no permutations of tasks in the schedule σ^{ord} between the machines $i \in \sigma^1$ and $i \in \sigma^2$ which lead to a decrease in the functional value.

Proof is based on Statement 1 and Corollary 1. Consider the schedule on machines $i_r \in \sigma^1$ and $i_s \in \sigma^2$. Suppose that a task j_k is processed on the machine i_r and a task j_p on the machine i_s . We have $l_{j_k} \leq l_{j_p}$ for these tasks, in accordance with the algorithm for the schedule σ^{ord} construction and due to the agreeability of weights and processing times. Let us swap these tasks, that is, let the task j_k be processed on the machine i_s and the task j_p on the machine i_r . As a result of such a permutation, in accordance with Corollary 1, the functional value increases. The processing times of tasks $j \in \sigma^2$ is greater than that of tasks in the sets P or S , according to Theorem 1. Therefore, according to Statement 1, tasks $j \in \sigma^2$ cannot be moved into the set P or the set S . Consequently, there are no permutations of tasks in the schedule σ^{ord} between schedules σ^1 and σ^2 which decrease the functional value. This proves the theorem.

Corollary 2. Optimization is only done in the schedule σ^1 . The schedule σ^2 is optimal by construction in accordance with Corollary 1.

Corollary 3. We check the sufficient signs of optimality on the schedule σ^1 and determine the estimate of deviation of the functional value from the optimum $\Omega_\Sigma(\sigma)$, which is equal to $\min(R_\Sigma(\sigma), \Delta_\Sigma(\sigma))$, on the schedule σ^1 , since the schedule σ^2 is optimal by construction.

Theorem 5. The functional value for WTPA problem is equal to the sum of the functional values of the schedules σ^1 and σ^2 .

Proof of Theorem 5 is obvious.

PSC-algorithm for WTPA problem solving. It has the following 8 steps.

1. Initial schedule construction by Algorithm A0.
2. Split the obtained schedule σ^{ord} into σ^1 and σ^2 where σ^1 is the schedule of tasks on the machines with $r_i < d$ and σ^2 is the schedule of tasks on the machines with $r_i \geq d$.
3. Execute the PSC-algorithm A1 or A2 [2] on the schedule σ^1 .
4. Analyze the obtained solution. If the polynomial component of the algorithm has been fulfilled, then the schedule σ^1 is optimal, go to step 5. Otherwise, go to step 6.
5. Determine the functional value for the schedule σ^1 . Go to step 7.

6. Determine the functional value and the estimate of deviation of the functional value from the optimum for the schedule σ^1 .
7. Determine the functional value for the schedule σ^2 which is optimal by construction.
8. Determine the functional value for WTPA problem in accordance with Theorem 5. Terminate.

Justification of the algorithm. As a result of the initial schedule construction for WTPA problem which is implemented by Algorithm A0, we obtain schedules σ^1 (the schedule on the machines with $r_i < d$) and σ^2 (the schedule on the machines with $r_i \geq d$). The functional value for the entire schedule σ is equal to the sum of the functional values for the schedules σ^1 and σ^2 . The schedule σ^2 is optimal by construction. But building and optimization of the schedule σ^1 is reduced to solving the TTP problem. Thus, PSC-algorithm for WTPA problem solving coincides with the PSC-algorithm for TTP problem solving on the schedule σ^1 . The polynomial component of the PSC-algorithm coincides with the polynomial approximation of the exact algorithm. As a result of the problem solving, we obtain either an exact optimal solution by the polynomial component of the algorithm (if at least one of the sufficient signs of optimality has fulfilled during its execution), or an approximate solution with an upper estimate of deviation of the functional value from the optimum which is $\Omega_\Sigma(\sigma)$.

The complexity of the polynomial component of the PSC-algorithm is determined by the complexity of the algorithm (A1 or A2) used to solve the problem. Their complexities are $O(n^2m)$ and $O(mn \log n)$, respectively.

Computational studies. To research the algorithm's efficiency, we used instances generator and solver written in C# in Microsoft Visual Studio 2010 environment. We randomly generated the task sets with uniform distribution of parameters. We chose processing times and weights of tasks from interval $[1, 200]$, then we assigned to each next task with the minimal processing time the next maximal of unassigned weights. The due date d was calculated as $0.7L/m$ where L is the sum of all processing times. The start times of machines were chosen from uniform distribution within $[0, 2d]$. We carried out 2,000 runs for each (n, m) pair. We tested problems with up to 40,000 tasks and 30 machines on a PC with 2 GBytes of RAM and a Pentium Core 2 Duo processor with 3.0 GHz frequency.

Average (for 100 runs) solving time appeared: from 1.53 ms at $n = 3,000$ and $m = 5$ to 2,025.14 ms at $n = 40,000$ and $m = 15$ for Algorithm A1; from 1.78 ms at $n = 3,000$ and $m = 30$ to 27.3 ms at $n = 40,000$ and $m = 20$ for Algorithm A2. The average frequency of obtaining an optimal solution was 90.3% for Algorithm A1 and 73.5% for Algorithm A2. The average deviation from an optimal solution was 0.000251 for Algorithm A1 and 0.000114 for Algorithm A2. After introducing additional types of permutations, Algorithm A2 becomes more efficient than Algorithm A1. In this case, we achieve an optimal functional value by the polynomial component of the algorithm for 92 % of instances.

Conclusions. We considered the problem of scheduling tasks on identical parallel machines. Weights and pro-

cessing times are agreeable, that is, a greater weight corresponds to a task with a shorter processing time. Additionally, arbitrary start times of the machines are given, they can be less than or greater than the due date, or to coincide with it. The problem in this formulation was solved for the first time. We presented an efficient PSC-algorithm for its solving, sufficient signs of optimality of obtained solutions, the estimate of deviation of an obtained solution from the optimum for each individual problem instance, and statistical studies of the PSC-algorithm which showed its high efficiency. We solved instances with dimensions up to 40,000 tasks with the number of machines up to 30. The average solving time by the algorithm that uses the most efficient types of permutations was 27.3 ms at this dimension. The average frequency of obtaining an optimal solution amounted to 90.3 %. The average deviation from an optimum was no more than 0.000251.

References

1. Yalaoui F. Minimizing total tardiness in parallel-machine scheduling with release dates. *Applied evolutionary computation*. 2012. Vol. 3, Iss. 1. P. 21–46. doi: 10.4018/jaec.2012010102
2. Zgurovsky M. Z., Pavlov A. A. *Combinatorial optimization problems in planning and decision making: theory and applications*. Cham: Springer, 2019. 526 p. doi: 10.1007/978-3-319-98977-8
3. Garey M. R., Johnson D. S. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: W. H. Freeman and Co., 1979. 348 p.
4. Lawler E. L. A “pseudopolynomial” algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*. 1977. Vol. 1. P. 331–342. doi: 10.1016/S0167-5060(08)70742-8
5. Pinedo M. L. *Scheduling: theory, algorithms, and systems*. 5-th ed. Cham: Springer, 2016. 690 p. doi: 10.1007/978-3-319-26580-3
6. Kramer A., Subramanian A. A unified heuristic and an annotated bibliography for a large class of earliness-tardiness scheduling problems. URL: <https://arxiv.org/abs/1509.02384> (accessed 05.05.2019).
7. Танаев В. С., Шкурба В. В. *Введение в теорию расписаний*. Москва: Наука, 1975. 256 с.
8. Kovalyov M. Y., Werner F. Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness. *Journal of Heuristics*. 2002, Vol. 8, Iss. 4. P. 415–428. doi: 10.1023/A:1015487829051
9. Згуровский М. З., Павлов А. А. *Труднорешаемые задачи комбинаторной оптимизации в планировании и принятии решений: монография*. Киев: Наукова думка, 2016. 716 с.
10. Pavlov A. A., Misura E. B., Melnikov O. V., Mukha I. P., Lishchuk K. I. Approximation algorithm for parallel machines total tardiness minimization problem for planning processes automation. *The Second International Conference on Computer Science, Engineering and Education Applications ICCSEEA 2019 (26–27 January 2019, Kiev)*. Cham: Springer Publ., 2020. pp. 459–467. doi: 10.1007/978-3-030-16621-2_43

References (transliterated)

1. Yalaoui F. Minimizing total tardiness in parallel-machine scheduling with release dates. *Applied evolutionary computation*. 2012, vol. 3, iss. 1, pp. 21–46. doi: 10.4018/jaec.2012010102
2. Zgurovsky M. Z., Pavlov A. A. *Combinatorial optimization problems in planning and decision making: theory and applications*. Cham, Springer Publ., 2019. 526 p. doi: 10.1007/978-3-319-98977-8
3. Garey M. R., Johnson D. S. *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco, W. H. Freeman and Co. Publ., 1979. 348 p.
4. Lawler E. L. A “pseudopolynomial” algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*. 1977, vol. 1, pp. 331–342. doi: 10.1016/S0167-5060(08)70742-8
5. Pinedo M. L. *Scheduling: theory, algorithms, and systems*. 5-th ed. Cham, Springer Publ., 2016. 690 p. doi: 10.1007/978-3-319-26580-3
6. Kramer A., Subramanian A. *A unified heuristic and an annotated bibliography for a large class of earliness-tardiness scheduling problems*. Available at: <https://arxiv.org/abs/1509.02384> (accessed 05.05.2019).
7. Tanaev V. S., Shkurba V. V.: *Vvedenie v teoriyu raspisaniy* [Introduction to scheduling theory]. Moscow, Nauka Publ., 1975. 256 p.
8. Kovalyov M. Y., Werner F. Approximation schemes for scheduling jobs with common due date on parallel machines to minimize total tardiness. *Journal of Heuristics*. 2002, vol. 8, iss. 4, pp. 415–428. doi: 10.1023/A:1015487829051
9. Zgurovsky M. Z., Pavlov A. A. *Trudnoreshaemye zadachi kombinatoroy optimizatsii v planirovani i prinyatii resheniy: monografiya* [Intractable problems of combinatorial optimization in planning and decision-making: monograph]. Kiev, Naukova dumka Publ., 2016. 716 p.
10. Pavlov A. A., Misura E. B., Melnikov O. V., Mukha I. P., Lishchuk K. I. Approximation algorithm for parallel machines total tardiness minimization problem for planning processes automation. *The Second International Conference on Computer Science, Engineering and Education Applications ICCSEEA 2019 (26–27 January 2019, Kiev)*. Cham, Springer Publ., 2020. pp. 459–467. doi: 10.1007/978-3-030-16621-2_43
11. Lawler E. L., Moore J. M. A functional equation and its application to resource allocation and sequencing problems. *Management Science*. 1969, vol. 16, no. 1, pp. 77–84. doi: 10.1287/mnsc.16.1.77
12. Yuan J. The NP-hardness of the single machine common due date weighted tardiness problem. *Journal of Systems Science and Complexity*. 1992, vol. 5, no. 4, pp. 328–333.

Received 07.04.2019

Відомості про авторів / Сведения об авторах / About the Authors

Павлов Олександр Анатолійович (Павлов Александр Анатольевич, Pavlov Alexander Anatolievich) – доктор технічних наук, професор, завідувач кафедри автоматизованих систем обробки інформації та управління, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»; м. Київ, Україна; ORCID: <http://orcid.org/0000-0002-6524-6410>; e-mail: pavlov.fiot@gmail.com

Місюра Олена Борисівна (Мисюра Елена Борисовна, Misura Elena Borisovna) – кандидат технічних наук, старший науковий співробітник НДІ інформаційних процесів Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»; м. Київ, Україна; ORCID: <http://orcid.org/0000-0002-4210-0996>; e-mail: elena_misura@ukr.net

Мельников Олег Валентинович (Мельников Олег Валентинович, Melnikov Oleg Valentinovich) – кандидат технічних наук, провідний інженер кафедри автоматизованих систем обробки інформації та управління Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»; м. Київ, Україна; ORCID: <http://orcid.org/0000-0002-2651-2739>; e-mail: oleg.v.melnikov72@gmail.com